NAD-2075 Series
Communications Appliance

User's Manual

Revision: 1.00

Portwell Inc.
3F, No. 92, Sec. 1, Nei-Hu Rd., Taipei 114, Taiwan, R.O.C.
Headquarter: +886-2-2799-2020 FAX: +886-2-2799-1010
http://www.portwell.com.tw
E M A I L :
I N F O @ M A I L . P O R T W E L L . C O M . T W

ISO 9001

Table of Contents

Chapter 1 Introduction 2

1.1 About This Manual 2

1.2 Manual Organization 2

1.3 Technical Support Information 2

Chapter 2 Get Started 3

2.1 Included Hardware 3

2.2 Before You Begin 3

2.3 The Chassis 4

2.4 Open the Chassis 4

2.5 Install or Remove a SODIMM 4

2.6 Remove and Install Battery 6

2.7 Install Compact Flash 6

2.8 Product Specifications 8

2.9 Hardware Configuration Setting 9

2.10 Use a Client Computer 15

2.11 BIOS Setup Information 16

2.12 Reset to Default Sample Code information 23

2.13 GPO Code information 30

2.14 WDT Sample Code information 30

Chapter 3 Operation Guide 34

3.1 Brief Guide of PPAP-2020 34

3.2 System Architecture 35

Chapter 1 Introduction

1.1 About This Manual

This manual describes all required information for setting up and using the NAD-2075 All mentioned below applies to the whole system, unless specially stated.

NAD-2075 provides the essential components for delivering optimal performance and functionality in the value communications appliance market segment. This manual should familiarize you with NAD-2075 operations and functions. NAD-2075 family has one, two or five on-board Ethernet ports to serve communication appliances, such as Firewall, which needs more Ethernet ports to connect external network (internet), demilitarized zone and internal network.

NAD-2075 features:

- Versatile networking and I/O capabilities: 1, 4 or 5 Ethernet ports
- One COM ports
- One miniPCI slot
- Onboard 256MB RAM, Up to 512 Mbytes or 1Gbytes of DDR2 memory

1.2 Manual Organization

The manual describes how to configure your NAD-2075 system to meet various operating requirements. It is divided into three chapters, with each chapter addressing a basic concept and operation of this whole system.

Chapter 1: Introduction. This section briefly talks about how this document is organized. It includes some guidelines for users who do not want to read through everything, but still helps you find what you need.

Chapter 2: Hardware Configuration Setting and Installation. This chapter shows how the hardware is put together, including detailed information. It shows the definitions and locations of Jumpers and Connectors that you can easily configure your system. Descriptions on how to properly mount the main memory are also included to help you get a safe installation. Reading this chapter will teach you how to set up NAD-2040.

Chapter 3: Operation Information. This section gives you illustrations and more information on the system architecture and how its performance can be maximized.

Any updates to this manual, technical clarification and answers to frequently asked questions would be posted on the web site: ***http://isc.portwell.com.tw***

1.3 Technical Support Information

Users may find helpful tips or related information on Portwell's web site: ***http://www.portwell.com.tw.*** A direct contact to Portwell's technical person is also available. For further support, users may also contact Portwell's headquarter in Taipei or your local distributors.

Taipei Office Phone Number: +886-2-27992020

Chapter 2 Get Started

This section describes how the hardware installation and system settings should be done.

2.1 Included Hardware

The following hardware is included in your kit:

- PPAP-2020 Communication Appliance System Board
- One null serial port cable

2.2 Before You Begin

To prevent damage to any system board, it is important to handle it with care. The following measures are generally sufficient to protect your equipment from static electricity discharge:

When handling the board, use a grounded wrist strap designed for static discharge elimination and touch a grounded metal object before removing the board from the antistatic bag. Handle the board by its edges only; do not touch its components, peripheral chips, memory modules or gold contacts.

When handling memory modules, avoid touching their pins or golden edge fingers. Put the value communications appliance system board and peripherals back into the antistatic bag when they are not in use or not installed in the chassis.

Some circuitry on the system board can continue operating even though the power is switched off. Under no circumstances should the Lithium coin cell be used to power the real-time clock be allowed to be shorted. The coin cell can heat under these conditions and present a burn hazard.

WARNING!

1. "CAUTION: Danger of explosion if battery is incorrectly replaced. Replace only with the same or equivalent type recommended by the manufacturer. Discard used batteries according to the manufacturer's instructions"
2. This guide is for technically qualified personnel who have experience installing and configuring system boards. Disconnect the system board power supply from its power source before you connect/disconnect cables or install/remove any system board components. Failure to do this can result in personnel injury or equipment damage.
3. Avoid short-circuiting the lithium battery; this can cause it to superheat and cause burns if touched.
4. Do not operate the processor without a thermal solution. Damage to the processor can occur in seconds.
5. Do not block air vents. Minimum 1/2-inch for clearance required.

2.3 The Chassis

The system is integrated in a customized chassis (***Fig. 2-1, Fig. 2-2***). On the front panel you will find the Power LED, Hard Disk LED and LAN LED. The back panel has Five LAN ports and a COM port.

Fig. 2-1 *Front view of the Chassis*

Fig. 2-2 *Back view of the Chassis*

2.4 Open the Chassis

1. Take off the four screws (three at the rear side and two at the right/left side and remove the top lead (***Fig. 2-3***).

Fig. 2-3 *Take off two screws*

2. The top lead (***Fig. 2-4***) can be removed from the base stand (***Fig. 2-5***).

Fig. 2-4 *The top lead*

Fig. 2-5 *The base stand*

2.5 Install or Remove a SODIMM

Follow these steps to upgrade or remove RAM module:

1. Install the system memory by pulling the socket's arm and pressing it into the slot gently. (***Fig. 2-6, 2-7***)

Fig. 2-6 *The memory slot*

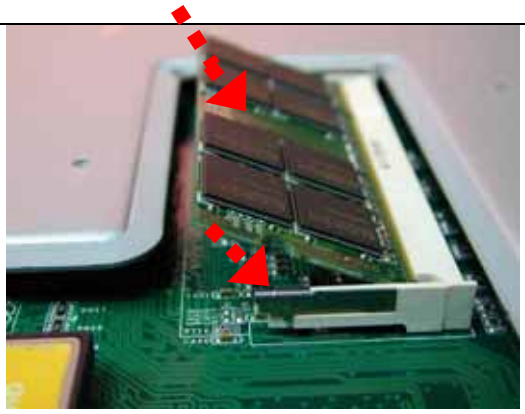


Fig. 2-7 *Install SODIMM*

2. Press down sodimm into slot (***Fig. 2-8***).

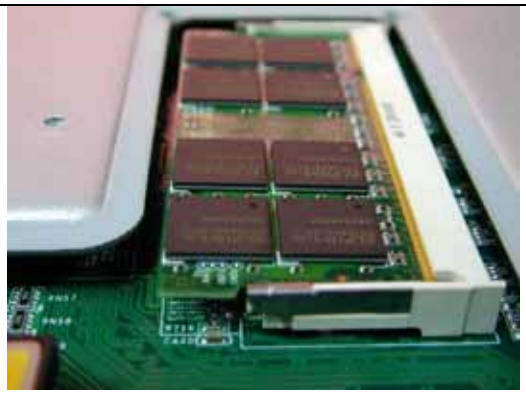
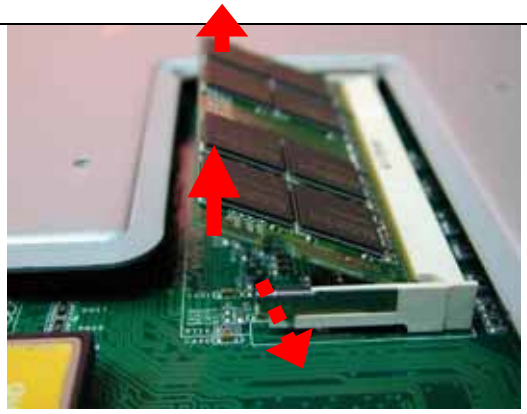


Fig. 2-8 *completes installs the memory*



Eject a SODIMM module

2.6 Remove and Install Battery

1. Press the metal clip back to eject the button battery (***Fig. 2-9***).
2. Replace it with a new one by pressing the battery with fingertip to restore the battery (***Fig. 2-10***).

Fig. 2-9 Eject *the battery*

Fig. 2-10 Restore *the battery*

2.7 Install Compact Flash

The system has an internal drive bay for one Compact Flash card drive. If the CF is not pre-installed, you can install it by yourself. Follow the steps below to install the CF:

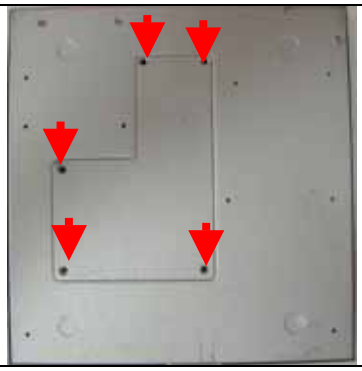
3. Fasten the five screws to lock bracket together (***Fig. 2-11a, 2-11b***).

Fig. 2-11a *Remove L type base under button case*

Fig. 2-11b Push *CF into the bracket*

4. Completion CF to the System Chassis (***Fig. 2-12***)

Fig. 2-12 completion *CF in system*



Fix all screws back (***Fig. 2-13***).

2.8 Product Specifications

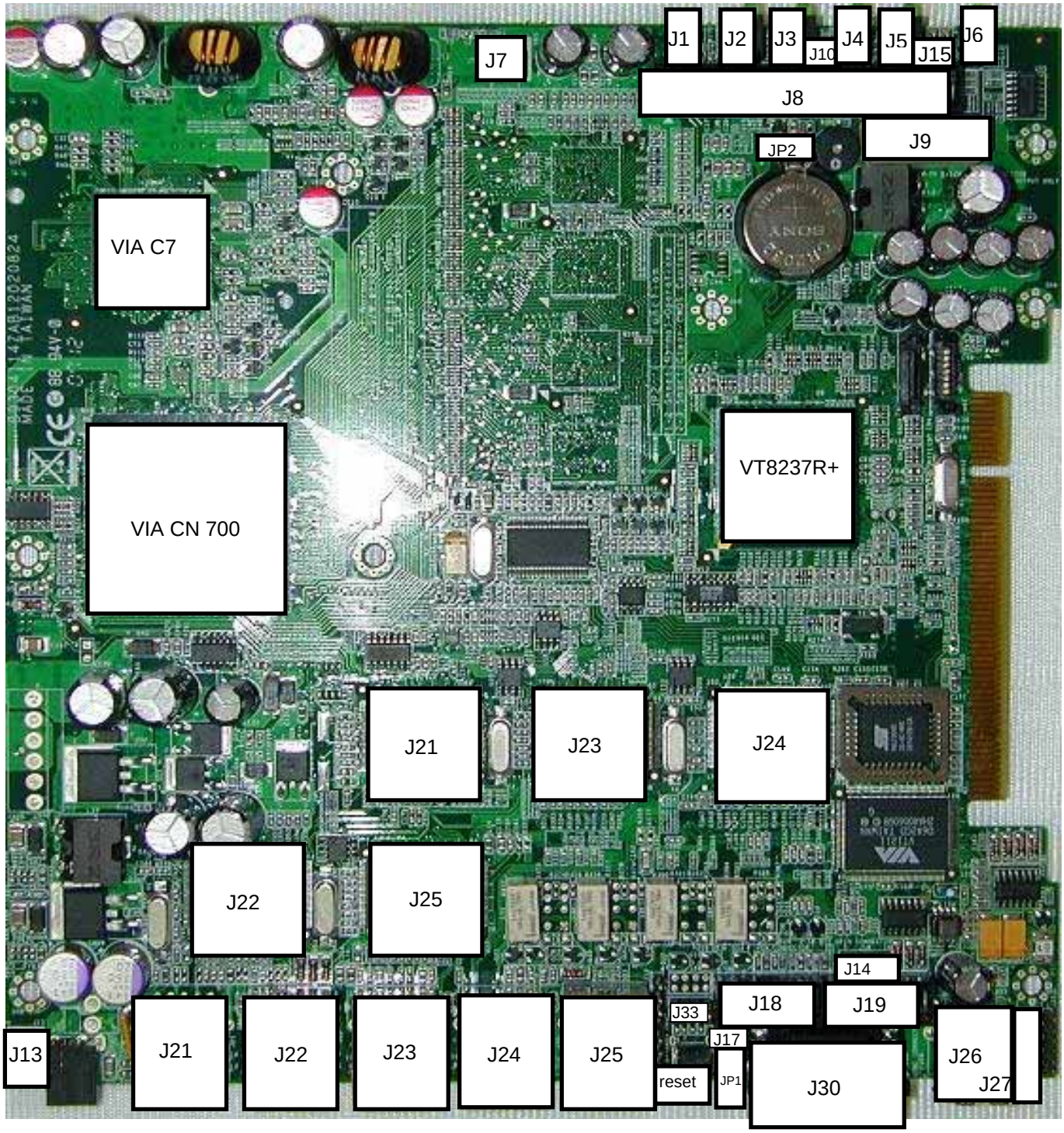
Model:	**NAD-2075**
Main Processor:	• VIA C7 processors (FSB 400/533Mhz)
BIOS:	• Award system BIOS with 512KB flash ROM to support DMI, PnP, APM function
Main Memory:	• Up to 1024MB 1.8V DDR2
L2 Cache Memory:	• 128KB 32-way built in (C7) CPU module
Chipset:	• VIA 8237R+
SATA Interface	• Two SATA I DMA133 Storage
PCI IDE Interface:	• One 40 Pin for DMA/33/66/100 IDE Storage
Serial Ports:	• Support two high-speed 16550 compatible UARTs with 16-byte T/R FIFOs
USB Interface:	• Support two USB2.0 ports for high speed I/O peripheral devices
Auxiliary I/O Interfaces:	• System reset switch, power okay LED, Ethernet activity LED, Ethernet speed LED, general purpose LED, alert LED and HDD LED interface
Power Input:	• Support one AC Adaptor with Adaptor input (power requirement: Input: 100-240V, Output: 15V == 4A)
On-board Ethernet:	• Five RealTek 8100C+ 10BASE-T/100BASE-TX Fast Ethernet controller with RJ-45 interface • Five RealTek 8110SC 32-bit Gb Ethernet controller with RJ-45 interface
Hardware Monitor:	Support on-board hardware monitor for • CPU fan, System fan • System voltages
Environmental Requirements:	• Operating Temperature: 5°C ~ 40°C • Storage Temperature: 0°C ~ 70°C • Relative Humidity: 5% ～ 95%, non-condensing
Dimension:	• 214mm(L) x 225mm (W) x 52mm (H)

2.9 Hardware Configuration Setting

This section gives the definitions and shows the positions of jumpers, headers and connectors. All of the configuration jumpers on PPAP-2011 are in the proper position. The default settings set by factory are marked with a star (★).

Jumpers

In general, jumpers on PPAP-2020 system board are used to select options for certain features. Some of the jumpers are configurable for system enhancement. The others are for testing purpose only and should not be altered. To select any option, cover the jumper cap over (Short) or remove (NC) it from the jumper pins according to the following instructions. Here NC stands for “Not Connected”.



PPAP-2020 Jumper Table

PPAP-2020 ZR4 jumper setting: (default setting:"*")

JP1:By-pass and normal mode

JP1 & J17 pin Header

+	-
6	5
4	3
2	1

JP1	Function
1-3 , 2-4 Short	Power on default at Normal mode , mode selection is S/W programmable★
3-5 , 2-4 Short	Power on default at Bypass mode , mode selection is S/W programmable
4-6 Short	Always at normal mode.

1-3,2-4 short

3-5,2-4 short

4-6 short

JP2: CMOS Clear

JP2	Function
1-2 Short	Normal Operation ★
2-3 Short	Clear CMOS Contents

J10: Reset to default function

J10	Function
1-2 Short	RESET TO DEFAULT
1-2 Open	Normal mode ★

Connector	Function	Remark
J1~J5	LAN LED	
J7	CPU FAN connector	
J8	IDE connector	
J9	+5V & +12V power connector(only output)	
J10	RESET TO DEFAULT	
J13	SYS FAN connector	
J14	8-bit GPO LED connector	
J15	HDD LED +Power LED	

J17	By-pass LED	
J18	VGA connector	
J19	COM2 connector	
J21~J24	RJ45 connector	
J26、J27	USB connector	
J30	COM1 connector	
J31	PICMG 1.0 connector	

Pin Assignments of Connectors

SW1: System reset

PIN No.	Signal Description
1	RST_SW
2	Ground

J7/J13: Fan power connector

PIN No.	Signal Description
1	GND
2	+12V
3	RPM signal

J14: 8-bit GPIO connector define

Pin	Signal Name	Pin	Signal Name
1	GPIO	**2**	GPIO
3	GPIO	**4**	GPIO
5	GPIO	**6**	GPIO
7	GPIO	**8**	GPIO
9	Ground	**10**	+5V

J15: HDD LED +Power LED connector define

Pin	Signal Name	Pin	Signal Name
1	+5V	**2**	-HD
3	+5V	**4**	-PWR

J18: VGA connector define

Pin	Signal Name	Pin	Signal Name
1	RED	**2**	DDCCLK
3	GREEN	**4**	Ground
5	BLUE	**6**	DDCDATA
7	HSYNC	**8**	Ground
9	VSYNC	**10**	N/C

J21/J22/J23/J24/J25: Ethernet5 RJ-45 interface connector

PIN No.	Signal Description
1	NC
2	LAN_MD0+
3	LAN_MD0-
4	TC1
5	NC
6	NC
7	TC2
8	LAN_MD1+
9	LAN_MD1-
10	NC
11	ACTIVE#
12	LINK#
13	LINK100#
14	+3.3V
15	Ground
16	Ground

J19: Serial port 2x5 shrouded connector (COM2)

PIN No.	Signal Description
1	DCD, Data carrier detect
2	DSR, Data set ready
3	RXD, Receive data
4	RTS, Request to send
5	TXD, Transmit data
6	CTS, Clear to send
7	DTR, Data terminal ready
8	RI, Ring indicator
9	GND, Ground
10	NC, Not connected

J8: IDE1 2x20 shrouded connector

PIN No.	Signal Description	PIN No.	Signal Description
1	Reset IDE	**21**	DRQ0
2	Ground	**22**	Ground
3	Host data 7	**23**	Host IOW
4	Host data 8	**24**	Ground
5	Host data 6	**25**	Host IOR
6	Host data 9	**26**	Ground
7	Host data 5	**27**	IOCHRDY0 {IOCHRDY1}
8	Host data 10	**28**	Not connected
9	Host data 4	**29**	DACK0 {DACK1}
10	Host data 11	**30**	Ground
11	Host data 3	**31**	IRQ14
12	Host data 12	**32**	Not connected
13	Host data 2	**33**	Address 1
14	Host data 13	**34**	Not connected
15	Host data 1	**35**	Address 0
16	Host data 14	**36**	Address 2
17	Host data 0	**37**	Chip select 0
18	Host data 15	**38**	Chip select 1
19	Ground	**39**	Activity
20	Not connected	**40**	Ground

J30: Serial port D-SUB9 connector (COM1)

PIN No.	Signal Description
1	Data Carrier Detect (DCD)
2	Receive Data (RXD)
3	Transmit Data (TXD)
4	Data Terminal Ready (DTR)
5	Ground (GND)
6	Data Set Ready (DSR)
7	Request to Send (RTS)
8	Clear to Send (CTS)
9	Ring Indicator (RI)

J26/J27: Dual USB port connector

PIN No.	Signal Description	PIN No.	Signal Description
1	+5V	**2**	N/C
3	USBD0-	**4**	Ground
5	USBD0+	**6**	USBD1+
7	Ground	**8**	USBD1-
9	N/C	**10**	+5V

J33:power off bypass function

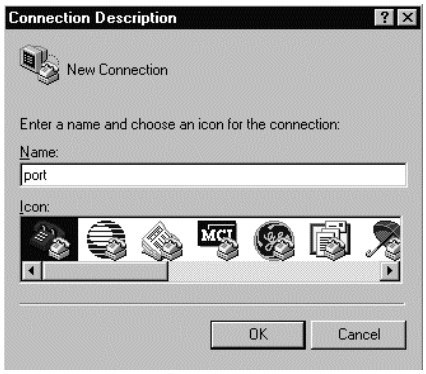
JP33	Function
1-5 , 2-6, 3-7 , 4-8 Short	Enable power off bypass function★
1-5 , 2-6, 3-7 , 4-8 Open	Disable power off bypass function★

Connection Using Hyper Terminal

If users use a headless NAD-2075, which has no mouse/keyboard and VGA output connected to it, the console may be used to communicate with NAD-2075.

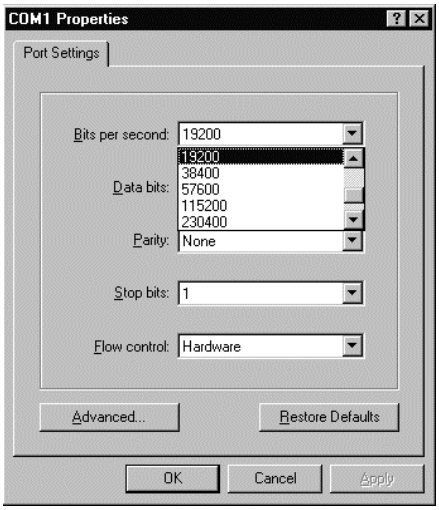
To access NAD-2075 via the console, Hyper Terminal is one of the choices. Follow the steps below for the setup:

1. Execute HyperTerminal under C:\Program Files\Accessories\HyperTerminal
2. Enter a name to create new dial

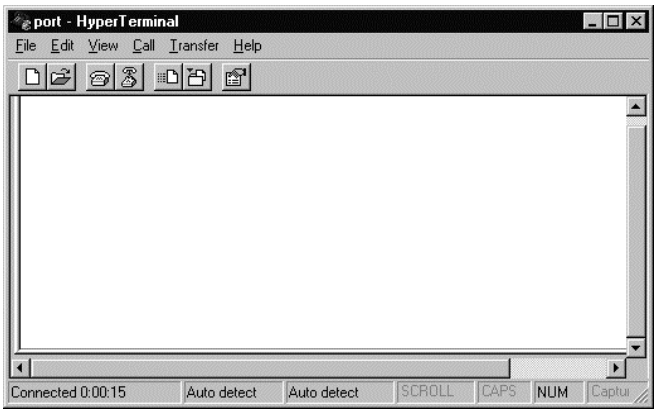


3. For the connection settings, make it Direct to COM1.

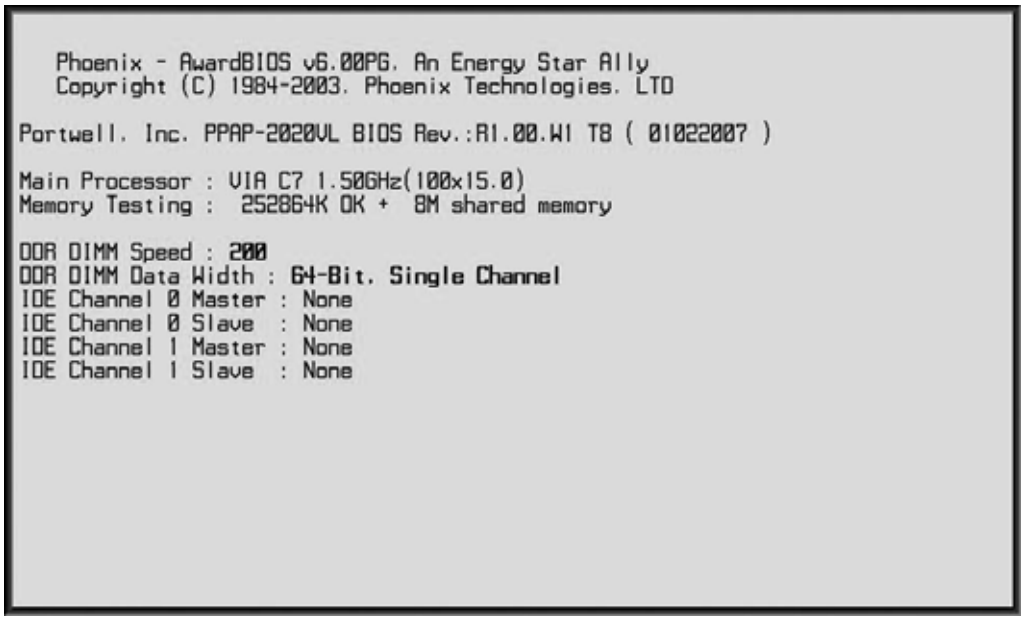
4. Please make the port settings to Baud rate 19200, Parity None, Data bits 8, Stop bits 1



5. Turn on the power of NAD-2075, after following screen was shown



6. You can then see the boot up information of NAD-2075


Phoenix - AwardBIOS v6.00PG. An Energy Star Ally
Copyright (C) 1984-2003. Phoenix Technologies. LTD
Portwell. Inc. PPAP-2020VL BIOS Rev.:R1.00.W1 T8 (01022007)
Main Processor : VIA C7 1.50GHz(100x15.0)
Memory Testing : 252864K OK + 8M shared memory
DDR DIMM Speed : 200
DDR DIMM Data Width : 64-Bit. Single Channel
IDE Channel 0 Master : None
IDE Channel 0 Slave : None
IDE Channel 1 Master : None
IDE Channel 1 Slave : None

When message “Hit <DEL> if you want to run Setup” appear during POST, after turning on or rebooting the computer, press **<Tab>** key ***immediately*** to enter BIOS setup program.

7. This is the end of this section. If the terminal did not port correctly, please check the previous steps.

2.11 BIOS Setup Information

NAD-2075 is equipped with the Award BIOS within Flash ROM. The BIOS has a built-in setup program that allows users to modify the basic system configuration easily. This type of information is stored in CMOS RAM so that it still retains during power-off periods. When system is turned on, NAD-2075 communicates with peripheral devices and checks its hardware resources against the configuration information stored in the CMOS memory. Whenever an error is detected, or the CMOS parameters need to be initially defined, the diagnostic program will prompt the user to enter the Setup program. Some errors are significant enough to abort the start-up.

Entering Setup

When you see the message “Hit <DEL> if you want to run Setup”, after turning on or rebooting the computer, press <Del> key ***immediately*** to enter BIOS setup program.

If you want to enter Setup but fail to respond before the message disappears, please restart the system either by first turning it off and followed by turning it on (COLD START) or simply press the "RESET" button. “WARM START” (press <Ctrl>, <Alt>, and <Delete> keys simultaneously) will do, too. Unless you press the keys at the right time, the system will not boot, an error message will display and you will be asked to do it again.

When no setting is stored in BIOS or the setting is missing, a message “Press <F1> to run Setup” will appear. Then press <F1> to run Setup or resume HIFLEX BIOS Setup. You can use the keyboard to choose among options or modify the system parameters to match the options with your system. The table shown on next page will show you all of keystroke functions in BIOS Setup.

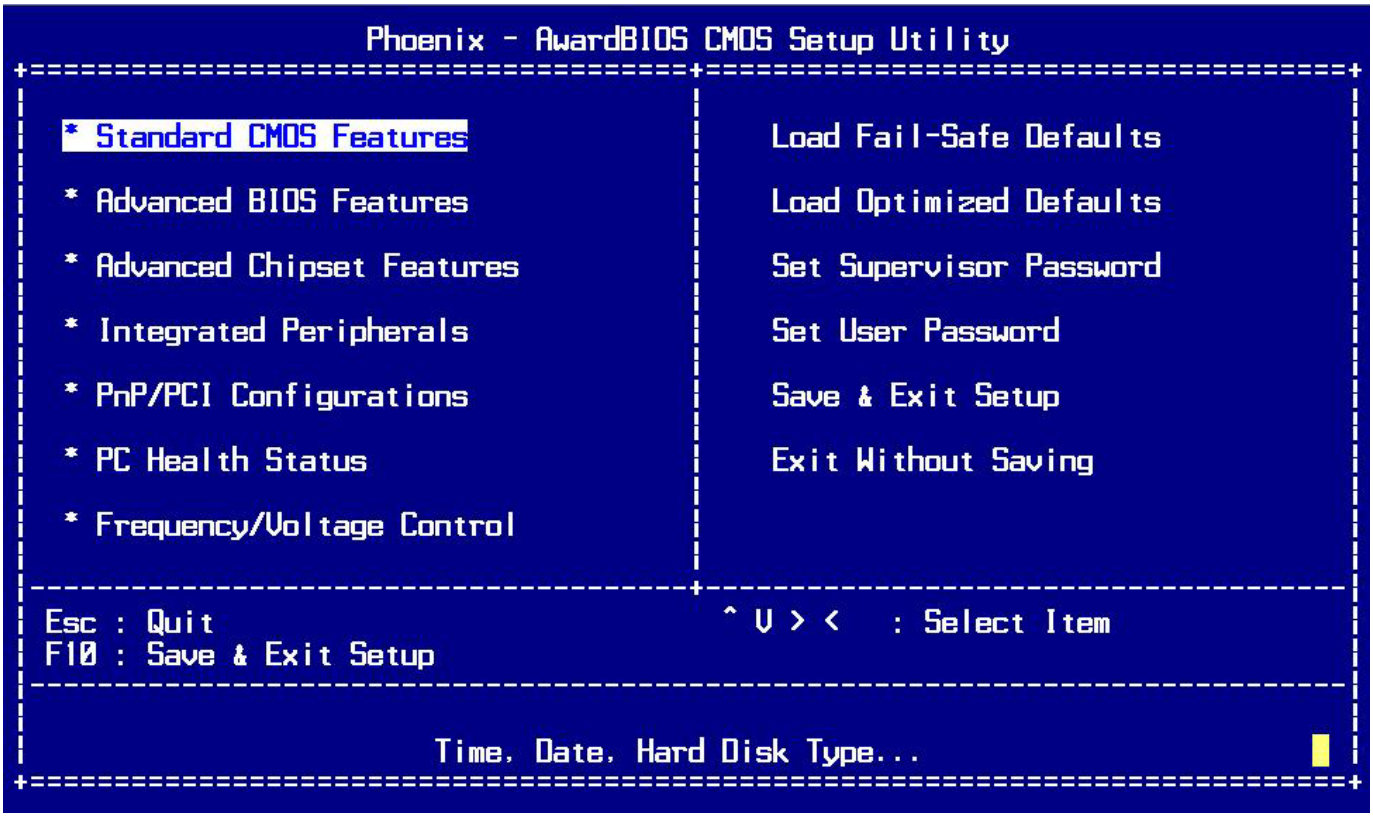
Keys to navigate within Setup menu

Key	Function
Up (↑)	Move to the previous item
Down (↓)	Move to the next item
Left (→)	Move to the item on the left (menu bar)
Right (←)	Move to the item on the right (menu bar)
Enter	Enter the item you desired
PgUp	Increase the numeric value or make changes
PgDn	Decrease the numeric value or make changes
＋	Increase the numeric value or make changes
－	Decrease the numeric value or make changes
Esc	**Main Menu:** Quit and not save changes into CMOS **Status Page Setup Menu and Option Page Setup Menu:** Exit current page and return to Main Menu
F1	General help on SETUP navigation keys
F5	Load previous values from CMOS
F6	Load the fail-safe defaults from BIOS default table
F7	Load the optimized defaults
F10	Save all the CMOS changes and exit

Main Menu

Once you enter NAD-2075 Award BIOS CMOS Setup utility, you should start with the Main Menu. The Main Menu allows you to select from eleven setup functions and two exit choices. Use arrow keys to switch among items and press <Enter> to accept or bring up the sub-menu.

Phoenix – Award BIOS CMOS Setup Utility



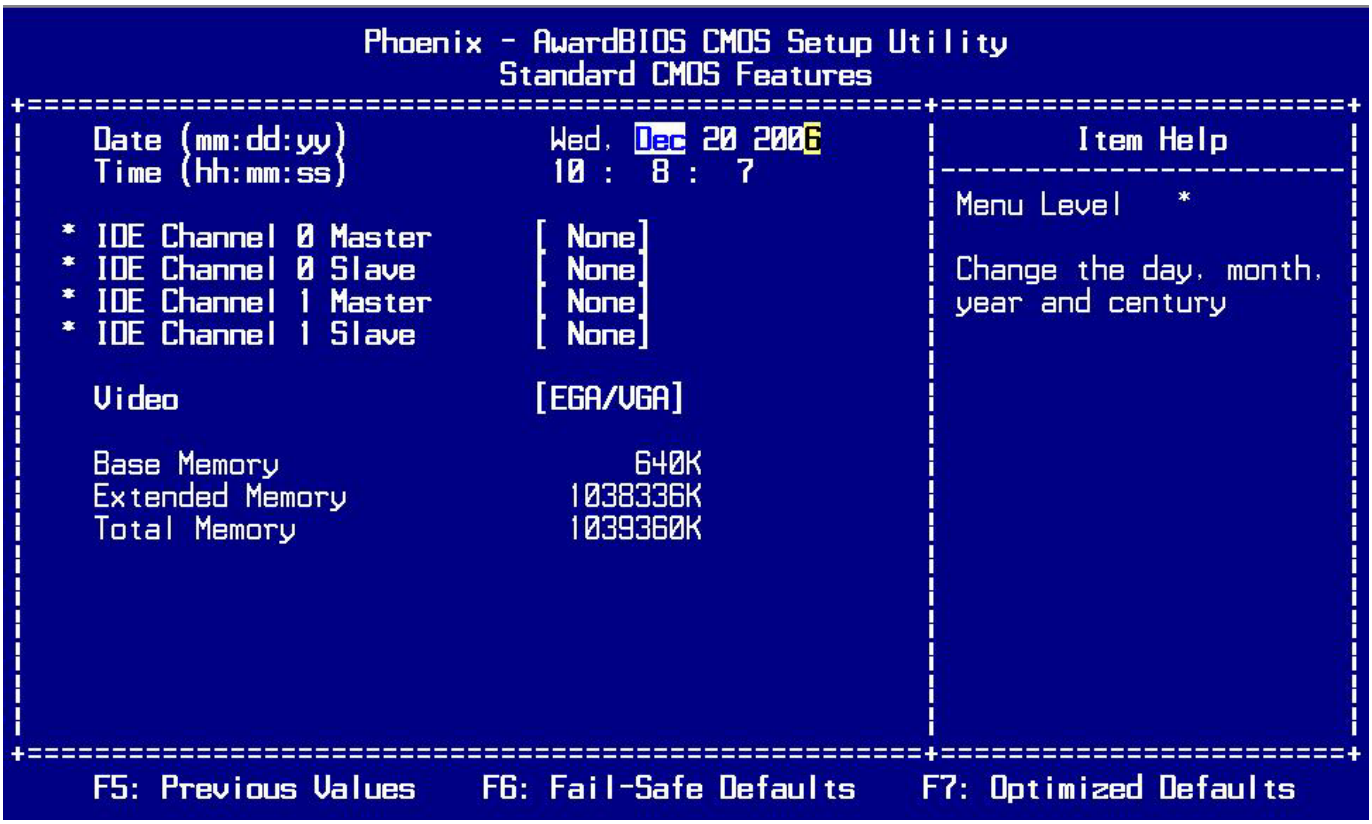
NOTE: It is strongly recommended to reload the optimized default setting if CMOS is lost or BIOS is updated.

Standard CMOS Setup Menu

This setup page includes all the items within standard compatible BIOS. Use the arrow keys to highlight the item and then use the <PgUp>/<PgDn> or <+>/<-> keys to select the value or number you want in each item and press <Enter> to certify it.

Follow command keys in CMOS Setup table to change *Date, Time, Drive type* and *Boot Sector Virus Protection Status*.

Screen Shot: Phoenix – Award BIOS CMOS Setup Utility


Phoenix - AwardBIOS CMOS Setup Utility
Standard CMOS Features
Date (mm:dd:yy) Wed, Dec 20 2006
Time (hh:mm:ss) 10 : 8 : 7
* IDE Channel 0 Master [None]
* IDE Channel 0 Slave [None]
* IDE Channel 1 Master [None]
* IDE Channel 1 Slave [None]
Video [EGA/VGA]
Base Memory 640K
Extended Memory 1038336K
Total Memory 1039360K
Item Help
Menu Level *
Change the day, month, year and century
F5: Previous Values
F6: Fail-Safe Defaults
F7: Optimized Defaults

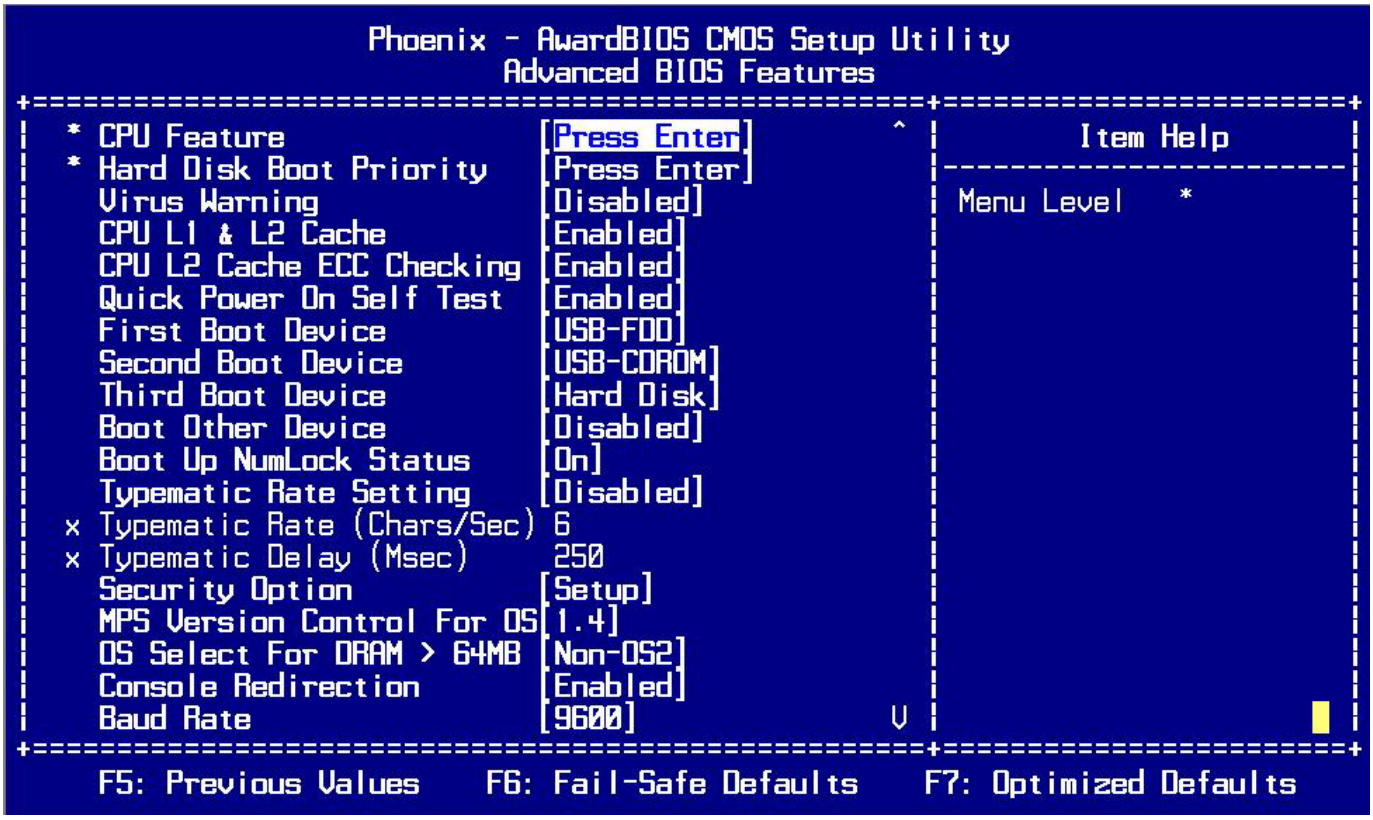
Menu Selections

Item	Options	Description
Date	mm:dd:yy	Set the system date. Note that the 'Day' automatically changes when you set the date
Time	hh:mm:ss	Set the system time
Video	EGA/VGA CGA 40 CGA 80 MONO	Select the default video device
Base Memory	N/A	Display the amount of conventional memory detected during boot up
Extended Memory	N/A	Display the amount of extended memory detected during boot-up
Total Memory	N/A	Display the total memory available in the system

Advance BIOS Features

This section allows user to configure your system for basic operation. Users will be able to select the system's default speed, boot-up sequence, keyboard operation, shadowing and security.

Screen Shot: Phoenix – Award BIOS CMOS Setup Utility


Phoenix - AwardBIOS CMOS Setup Utility
Advanced BIOS Features
* CPU Feature [Press Enter]
* Hard Disk Boot Priority [Press Enter]
Virus Warning [Disabled]
CPU L1 & L2 Cache [Enabled]
CPU L2 Cache ECC Checking [Enabled]
Quick Power On Self Test [Enabled]
First Boot Device [USB-FDD]
Second Boot Device [USB-CDROM]
Third Boot Device [Hard Disk]
Boot Other Device [Disabled]
Boot Up NumLock Status [On]
Typematic Rate Setting [Disabled]
x Typematic Rate (Chars/Sec) 6
x Typematic Delay (Msec) 250
Security Option [Setup]
MPS Version Control For OS [1.4]
OS Select For DRAM > 64MB [Non-OS2]
Console Redirection [Enabled]
Baud Rate [9600]
Item Help
Menu Level *
F5: Previous Values
F6: Fail-Safe Defaults
F7: Optimized Defaults

Internal Cache/External Cache

These two categories speed up memory access. However, it depends on CPU/chipset

design.

Enabled	Enable cache
Disabled	Disable cache

Quick Power On Self Test

This category speeds up Power On Self Test (POST) after you power up the computer. If it is set to Enable, BIOS will shorten or skip some check items during POST.

Enabled	Enable quick POST
Disabled	Normal POST

Boot Up NumLock Status

Select power on state for NumLock.

The choice: Enabled/Disabled.

Gate A20 Option

This entry allows user to select how the gate A20 is handled. The gate A20 is a device used to address memory over 1 Mbytes. Originally, the gate A20 was handled via a pin on the keyboard. But now, though keyboards still provide this support, it is more common, and much faster, for the system chipset to provide support for gate A20.

Normal	Keyboard
Fast	Chipset

Typematic Rate Setting

Keystrokes repeat at a rate determined by the keyboard controller. When enabled, the typematic rate and typematic delay can be selected.

The choice: Enabled/Disabled.

Typematic Rate (Chars/Sec)

Set the how many number of times a second to repeat a keystroke when a key is holding down.

The choice: 6, 8, 10, 12, 15, 20, 24 and 30.

Typematic Delay (Msec)

Set the delay time after the key is held down before it begins to repeat the keystroke.

The choice: 250, 500, 750 and 1000.

Security Option

Select whether the password is required every time the system boots or only when you enter setup.

System	The system will not boot and access to Setup will be denied if the correct password is not entered at the prompt.
Setup	The system will boot and access to Setup will be denied if the correct password is not entered at the prompt.

Note: To disable security, select PASSWORD SETTING at Main Menu and then user will be asked to enter password. Do not type anything and simply press <Enter>, it will disable security. Once the security is disabled, the system will boot up and user can enter Setup freely.

OS Select for DRAM > 64MB

Select the operating system that is running with more than 64MB of RAM on the system.

The choice: Non-OS2, OS2.

Console Redirection

Set the UNIX Console redirect to the terminal from COM1**.**

The choice: Enabled/Disabled.

Baud Rate

Set the RS-232 baud rate speed.

The choice: 9600, 19200, 38400, 57600 and 115200.

Advanced Chipset Features

This section allows user to configure your system for AT clock, DRAM timings...

Onboard LAN BootROM

Phoenix - AwardBIOS v6.00PG. An Energy Star Ally
Copyright (C) 1984-2003. Phoenix Technologies. LTD
Proventia BIOS GX3002 Version R1.00.W1 T8 (11212006)
Main Processor : VIA C7 1.00GHz(100x10.0)
Memory Testing : 835648K OK + 8M shared memory
DDR DIMM Speed : 533
DDR DIMM Data Width : 64-Bit. Single Channel
IDE Channel 0 Master : None
IDE Channel 0 Slave : None
IDE Channel 1 Master : None
IDE Channel 1 Slave : None
Press TAB to enter SETUP by Terminal
Press L to Boot From LAN. or press any other key to boot normally

User can press “L” for boot from LAN.

2.12 Reset to Default Sample Code information

```
; For  PPAP-2020, RESET to Default  testing
; By  Frank Hsu ,  12/22/2004
;       Reset to default status  can be read from  Via VT8235_GPI24
;
;               After Power On reset, GPI24 =high( 1 ) , normal state
;    If Reset to Default (RST2DF) Button,J14, pressed ( Triggered )
;     ,then GPI24  will  be  latch  to low ( 0 ).
;
;    RST2DF  register  can  be  set to high by VT8235_GPO31.
;    Write  a  pulse  timing ( High1_low_high2 )  to  set  RST2DF to 1.
;    High1 : output  GPO31  high , and  keep  30 us.
;    Low   : output  GPO31  low  , and  keep  30 us.
;    High2 : output  GPO31  high  again , and  keep  high  always.
;
;
;  Programming Guide :
;  PG_Step1 :  Set VT8235_Pin_AE5  to  GPI24 (B0:D17:F0_Offset_E6h_Bit0P0)
;           Set VT8235_Pin_AC6  to  GPO31 (B0:D17:F0_Offset_E6h_Bit7P1)
;           Let GPI24  non_invert --> (B0:D17:F0_Offset_E0h_Bit4P0)
;           Let GPI31  TTL output --> (B0:D17:F0_Offset_E7h_Bit7P1)
;  PG_Step2 :  Get PM_IO_BASE  from B0:D17:F0_Offset_[89h..88h]
;  PG_Step3 :  Get GPI24 status from bit0 of IOPORT ( PM_IO_BASE + 4Bh )
;           Output GPO31 to bit7 of IOPORT ( PM_IO_BASE + 4Fh )

        .MODEL small
        .386
        .STACK 200h
```

```
        .data

PROMP1 DB'PORTWELL PPAP-2011,2011RSTD.exe, V1.00 12-22-2004,All rights reserved.$'


PROMP1_1 DB '  For  PPAP-2011   Reset-to-Default test .',13,10,'$'
PROMP_2_CR_LF  db  0Dh, 0Ah,0Dh, 0Ah, '$'
PROMP_Str1 db '  Reset-To-Default status latched by a F/F. ',0dh,0ah,'$'
PROMP_Str2 db '      This status bit = 1 ---> Normal.      ',0dh,0ah,'$'
PROMP_Str3 db '      This status bit = 0 ---> RST2DF button,J14,has been pressed.',0dh,0ah,'$'
PROMP_Str4 db '      This status bit can be read by VT8235_GPI24, ',0dh,0ah,'$'
PROMP_Str5 db '        and can be cleared by an VT8235_GPO31  High1-Low-High2 pulse.',0dh,0ah,'$'
PROMP_Str6 db '                                          ',0dh,0ah,'$'
PROMP_Str7 db '        High1 =   30us  High  level          ',0dh,0ah,'$'
PROMP_Str8 db '        Low   =   30us  Low   level         ',0dh,0ah,'$'
PROMP_Str9 db '        High2 =   High  level again and no level change from now on.',0dh,0ah,'$'
PROMP_StrA db '                                          ',0dh,0ah,'$'
PROMP_rst2df db 0dh,0ah,' Press the Reset-to-Default button and then release it for the test NOW!$'
PROMP_anykey db 0dh,0ah,' Ready ? If yes , then Press any key to start test ..... $'

PROMP_err1 db 0dh,0ah,' ***** "Reset-to-Default F/F Initialization"  Failed. *****',0dh,0ah,'$'
PROMP_err1_1 db ' ( This may be a H/W error or Reset-to-Default button has ever been
pressed ! )',0dh,0ah,'$'
PROMP_err2 db 0dh,0ah,' ***** "Reset-to-Default event latched by F/F "  Failed. *****',0dh,0ah,'$'
PROMP_err3 db 0dh,0ah,' ***** "Clear Reset-to-Default F/F status "  Failed. *****',0dh,0ah,'$'
PROMP_TEST_OK db ' <<..... PPAP-2011 RESET-TO-DEFAULT test OK ..... ^_^... >>',0dh,0ah,'$'
PROMP_TEST_fail db ' <<*****  PPAP-2011 RESET-TO-DEFAULT test FAIL ***** !!! >>',0dh,0ah,'$'
PROMP_Qkey db 0dh,0ah,'Press "Q" key to stop test and return to DOS; or other key to go on next test.$'
debugtesting db 0dh,0ah,' <<<<< This is for debugging  only >>>>> ',0dh,0ah,'$'

ERROR_CODE            db ?    ;
PM_IO_BASE            dw ?    ;
        .code
programstart:
        mov ax,@data
        mov ds,ax

        lea dx,PROMP_2_CR_LF
        mov ah,09h
        int 21h

        lea dx,PROMP1
        mov ah,09h
        int 21h

        lea dx,PROMP_2_CR_LF
        mov ah,09h
        int 21h

        lea dx,PROMP1_1
        mov ah,09h
        int 21h

        lea dx,PROMP_2_CR_LF
        mov ah,09h
        int 21h

        lea dx,PROMP_Str1
        mov ah,09h
        int 21h
        lea dx,PROMP_Str2
        mov ah,09h
        int 21h
        lea dx,PROMP_Str3
        mov ah,09h
        int 21h
```

```
        lea dx,PROMP_Str4
        mov ah,09h
        int 21h
        lea dx,PROMP_Str5
        mov ah,09h
        int 21h
        lea dx,PROMP_Str6
        mov ah,09h
        int 21h
        lea dx,PROMP_Str7
        mov ah,09h
        int 21h
        lea dx,PROMP_Str8
        mov ah,09h
        int 21h
        lea dx,PROMP_Str9
        mov ah,09h
        int 21h
        lea dx,PROMP_StrA
        mov ah,09h
        int 21h


        mov ERROR_CODE,00h   ; Init  Error code to 00

;  PG_Step1 :  Set VT8235_Pin_AE5  to  GPI24 (B0:D17:F0_Offset_E6h_Bit0P0)
;              Set VT8235_Pin_AC6  to  GPO31 (B0:D17:F0_Offset_E6h_Bit7P1)
;              Let GPI24  non_invert --> (B0:D17:F0_Offset_E0h_Bit4P0)
;              Let GPI31  TTL output --> (B0:D17:F0_Offset_E7h_Bit7P1)

        mov  dx,0CF8h      ; PCI Config Read
        mov  eax,800088E4h ; B0:D31:F0:Offset_E4h
        out  dx,eax
        mov  dx,0CFCh
        in   eax,dx

        ror  eax,10h       ; swap [E5..E4]  with [E7..E6]
        or   ax,8080h      ; E7_Bit7P1 , E6_Bit7P1
        and  ax,0FFFEh     ; E6_Bit0P0
        rol  eax,10h       ; swap back

        out  dx,eax        ; Update these 3 bits


        mov  dx,0CF8h      ; PCI Config Read
        mov  eax,800088E0h ; B0:D31:F0:Offset_E0h
        out  dx,eax
        mov  dx,0CFCh
        in   eax,dx

        and  al,0EFh       ; E0_Bit4P0


        out  dx,eax        ; Update this bit

;  PG_Step2 :  Get PM_IO_BASE  from B0:D17:F0_Offset_[89h..88h]

        mov  dx,0CF8h      ; Get PM_IO_BASE
        mov  eax,80008888h ; B0:D17:F0:Offset_88h
        out  dx,eax
        mov  dx,0CFCh
        in   eax,dx

        and  al,0feh       ; bit0 cleared to 0.

        mov  PM_IO_BASE,ax
```

```
;  PG_Step3 :  Get GPI24 status from bit0 of IOPORT ( PM_IO_BASE + 4Bh )
;              Output GPO31 to bit7 of IOPORT ( PM_IO_BASE + 4Fh )


;  Testing  way :
;  --- t1
;  Read GPI24 first , GPI24=1  ? if yes,pass ; if no, failed
;
;  --- t2
;  RST2DF button pressed and released , read GPI24 ,GPI24 = 0 ? if yes, pass ; if no, failed
;
;  --- t3
;  Set RST2DF status to 1 ,read GPI24 ,GPI24 = 1 ? if yes, pass ; if no, failed
;------------------------------------------------------------------- t_start

;  ---- t1 start
;  GPI24 , read its  status , initialization  will  be  1 .
;
;            How  to  read  GPI24
;            ---------------------------------------------------
;            Get GPI24 status from bit0 of ( PM_IO_BASE+4Bh)
;              0 = low ( REset to DF ), 1= high level ( Normal )
;            ---------------------------------------------------

        mov  dx,PM_IO_BASE
        add  dx,4Bh
        in   al,dx

        BT   ax,0
        jc   okay_test1   ; Carry --> GPI24 =1

    ; no , error  message  display

         lea dx,promp_err1
         mov ah,09h
         int 21h
         lea dx,promp_err1_1
         mov ah,09h
         int 21h

         or   ERROR_CODE, 01   ;  Error code bit0_P1

         call  KB_Wait

;  ---- t1 end

         okay_test1 :

;  ---- t2 start
         lea dx,promp_rst2df
         mov ah,09h
         int 21h

         lea dx,promp_anykey
         mov ah,09h
         int 21h


         xor al,al          ; halt for ready?  Any  key  pressed to go on.
 WAIT_KB_0:
         mov ah,1
         int 21h

         cmp al,0
         je  WAIT_KB_0
```

```
        lea dx,PROMP_2_CR_LF
        mov ah,09h
        int 21h


;  test  RST2DF  button  pressed


       mov  dx,PM_IO_BASE
       add  dx,4Bh
       in   al,dx

       BT   ax,0
       jnc   okay_test2  ; No Carry --> GPI24 =0 , RSTDF  pressed
                  ; test okay, go on next test.

    ; Carry --> test failure --> error  message  display

        lea dx,promp_err2
        mov ah,09h
        int 21h

        or  ERROR_CODE,02h   ; ERROR_CODE_Bit1_P1

        call  KB_Wait

;  ---- t2 end

        okay_test2 :

;  ---- t3 start    ,Clear  RST2DF  F/F
;  GPO31  write 1,0,1
;

   ; =========   Write GPO31 1-0-1   start


        call  IODELAY ;  io delay
        mov   dx,PM_IO_BASE
        add   dx,4Fh
        in    al,dx              ;  Read  the  byte  data first

        or    al,80h              ; GPO31_P1 first
        out   dx,al

       call   FIXDELAY             ;  30 us delay

        and  al,7Fh               ; output GPO31_P0 then
        out  dx,al

       call   FIXDELAY             ;  30 us delay

        or    al,80h              ; GPO31_P1 Finally
        out   dx,al


   ; =========   Write GPO31 1-0-1   end

       mov  dx,PM_IO_BASE
       add  dx,4Bh
       in   al,dx

       BT   ax,0
       jc   okay_test3   ; Carry --> GPI24 =1
```

```
    ; no , error  message  display

        lea dx,promp_err3
        mov ah,09h
        int 21h

        or   ERROR_CODE,04h

;  ---- t3 end

   okay_test3 :

        cmp  ERROR_CODE,00h
        je  test_ok

 test_fail :

        lea dx,PROMP_2_CR_LF
        mov ah,09h
        int 21h

        lea dx,promp_TEST_fail
        mov ah,09h
        int 21h

        jmp return_to_dos

   test_ok :
        lea dx,promp_TEST_OK
        mov ah,09h
        int 21h

return_to_dos :
        mov  ah,4ch  ; Return  to  DOS
        int  21h

;------------------------------------------------------------------- t_end


IODELAY  PROC    near
        push  ax
        push  dx

        mov   dx,0edh
        in    al,dx
        jmp   $+2
        mov   dx,0edh
        in    al,dx

        pop   dx
        pop   ax
        ret
IODELAY  ENDP


KB_wait  PROC    near

        push ax
        push bx
        push cx
        push dx

        lea dx,PROMP_Qkey
        mov ah,9                ; Display "Q" key prompt
```

```
        int 21h

        xor al,al
WAIT_KB:
        mov ah,1
        int 21h

        cmp al,0
        je  WAIT_KB

        cmp al,51h   ; "Q"  pressed ?
        je  test_fail
        cmp al,71h   ; "q"  pressed ?
        jne call_return
        jmp test_fail  ;

call_return :

        lea dx,PROMP_2_CR_LF
        mov ah,09h
        int 21h

        pop dx
        pop cx
        pop bx
        pop ax

        ret

KB_wait  ENDP



;-------------------------------------------------------------------;
;                       FIXED_DELAY                                 ;
;-------------------------------------------------------------------;
;       Input : (CX) count of 15 microseconds to wait               ;
;               STACK PRESENT                                       ;
;       Output: NONE                                                ;
;       CX=2 , 15us x 2 = 30 us                                     ;
;                                                                   ;
; This routine is called to wait for 15 microseconds * count in     ;
; (CX), then return.  Gives a programmed software delay.            ;
;-------------------------------------------------------------------;
FIXDELAY  PROC    near
        push cx
        push dx
        push ax
        pushf

        mov  cx,02h

       mov     dx,61h
       in      al,dx               ;
       jmp     $+2
       jmp     $+2
       and     al,00010000b                ;
       mov     ah,al                       ;
fixed_delay_1:
       in      al,dx               ;
       jmp $+2
       jmp $+2
       and     al,00010000b                ;
       cmp     al,ah                       ;
       jz      short fixed_delay_1         ;
       mov     ah,al                       ;
```

```
        loop    short fixed_delay_1         ;

        popf
        pop  ax                         ;
        pop  dx
        pop  cx

        ret
 FIXDELAY  ENDP


END programstart
```

2.13 GPO Code information

```
/* This Program is to test PPAP-2020 GP0O Port0-7        */
/* Programer: John Ma                                    */
#include <stdio.h>
#include <conio.h>
#include <dos.h>

#define GPIO_Base          0x404d
#define d_time             2000


int main()
{
  union REGS regsi;
  unsigned char i;

  printf("\n\n PPAP-2011 GPO TEST PROGRAM R1.00");

  /* Test each GPIO */
  outportb(GPIO_Base, 0x55);        /* make sure read data from GPIO pin */
  delay(d_time);
  outportb(GPIO_Base, 0xaa);        /* make sure read data from GPIO pin */
  delay(d_time);
  outportb(GPIO_Base, 0x0f);        /* make sure read data from GPIO pin */
  delay(d_time);
  outportb(GPIO_Base, 0xf0);        /* make sure read data from GPIO pin */
  delay(d_time);
  outportb(GPIO_Base, 0x00);        /* make sure read data from GPIO pin */
  printf("\nTEST Finish!!");
  return(0);
}
```

2.14 WDT Sample Code information

```
`/*             W83697 WDT DEMO PROGRAM
                File Name : 697_WDT.C
                Write by : John Ma
*/

#include <stdio.h>
#include <conio.h>
```

```
#include <dos.h>

unsigned char time_out;                /* recode WDT time-out value */

void show_title(void)
{
   clrscr();
   printf("\n\nPortwell Inc. W83697HF WDT DEMO PROGRAM V1.00\n\n");
   printf("1.Set WDT Time-out value and Enable WDT\n");
   printf("2.Retriggle WDT\n");
   printf("3.Disable WDT and Exit\n");
   printf("\n\nPress 1 - 3 :");
}

void enable_WDT(void)
{
   printf("\n\nPress number 1 - 255 to select time-out time(second)");
   scanf("%d",&time_out);
   asm cli;
   /* Step1. Enter W83697 extended function mode:
      write 0x87 to port 2e twice
   */
   outportb(0x2e, 0x87);
   outportb(0x2e, 0x87);
   /* Step2. Select W83697 Pin119 to be WDTO#
      write 0x29 to port 2e to point W83697 CR_29
      write 0x20(bit6:5) to port 2f to select WDTO#
   */
   outportb(0x2e, 0x29);
   outportb(0x2f, 0x20);
   /* Step3. Select W83697 logic device 8:
      write 0x07 to port 2e to point W83697 CR_07
      write 0x08 to port 2f to select logic device 8
   */
   outportb(0x2e, 0x07);
   outportb(0x2f, 0x08);
   /* Step4. Enable Logic device 8:
      write 0x30 to port 2e to point W83697 CR_30
      write 0x1 to port 2f to enable
   */
   outportb(0x2e, 0x30);
   outportb(0x2f, 0x1);
   /* Step5. Config W83697 WDT using second to be unit:
      write 0xf3 to port 2e to point W83977ATF CR_f3
      write 0x00 to port 2f to select time-out unit is second
   */
   outportb(0x2e, 0xf3);
   outportb(0x2f, 0x00);
   /* Step6. Set WDT time-out time:
      write 0xf4 to port 2e to point W83697 CR_f4
      write time_out to port 2f to set time-out time
   */
   outportb(0x2e, 0xf4);
   outportb(0x2f, time_out);
   /* Step7. Exit W83697 extended function mode:
      write 0xaa to port 2e
   */
   outportb(0x2e, 0xaa);
   asm sti;
}

void retriggle_WDT(void)
{
   asm cli;
   /* Step1. Enter W83697 extended function mode:
      write 0x87 to port 2e twice
```

```
    */
    outportb(0x2e, 0x87);
    outportb(0x2e, 0x87);
    /* Step2. Select W83697 logic device 8:
       write 0x07 to port 2e to point W83697 CR_07
       write 0x08 to port 2f to select logic device 8
    */
    outportb(0x2e, 0x07);
    outportb(0x2f, 0x08);
    /* Step3. Retriggle WDT time-out time:
       write 0xf4 to port 2e to point W83697 CR_f4
       write time_out to port 2f to set time-out time in W83697
    */
    outportb(0x2e, 0xf4);
    outportb(0x2f, time_out);
    /* Step4. Exit W83697 extended function mode:
       write 0xaa to port 2e
    */
    outportb(0x2e, 0xaa);
    asm sti;
    printf("\n\n!!! Trigle !!!");
    delay(1000);
}

void disable_WDT(void)
{
    asm cli;
    /* Step1. Enter W83697 extended function mode:
       write 0x87 to port 2e twice
    */
    outportb(0x2e, 0x87);
    outportb(0x2e, 0x87);
    /* Step2. Select W83697 logic device 8:
       write 0x07 to port 2e to point W83697 CR_07
       write 0x08 to port 2f to select logic device 8
    */
    outportb(0x2e, 0x07);
    outportb(0x2f, 0x08);
    /* Step3. Disable WDT :
       write 0xf4 to port 2e to point W83697 CR_f4
       write 0x00 to port 2f to set time-out time in W83697
    */
    outportb(0x2e, 0xf4);
    outportb(0x2f, 0x00);
    /* Step4. Exit W83697 extended function mode:
       write 0xaa to port 2e
    */
    outportb(0x2e, 0xaa);
    asm sti;
}

int main(void)
{
    int key;

    show_title();
    while(1)
    {
     key = getch();
     switch (key)
     {
     case '1':
         enable_WDT();
         break;
     case '2':
         retriggle_WDT();
```

```
            break;
        case '3':
            disable_WDT();
            return(0);
        default:
            break;
        }
        show_title();
        }
}
```

Chapter 3 Operation Guide

3.1 Brief Guide of PPAP-2020

PPAP-2020 is a Communication Appliance computing board based on VIA VT8237R+ chipset technology. PPAP-2020 has four/five on-board LAN ports to serve communication appliances, such as Firewall, which needs four Ethernet ports to connect external network (internet), demilitarized zone and internal network. Different I/O management policies can be applied respectively to individual network to achieve the highest security level. The target market segment is communication appliance including Virtual Private Network, Load Balancing, Quality of Service, Intrusion Detection, Virus Detection, Firewall and Voice Over IP.

This PPAP-2020 system board is eligible with VIA Eden processor EBGA package (Eden Esp8000) and On-board 256Mb or higher DDRAM. The enhanced on-board PCI IDE interface supports 1 drive up to PIO mode 4 timing and Ultra DMA/100 synchronous mode feature. The on-board super I/O chipset integrates two serial ports driven by two high performance 16550C-compatible UARTs to provide 16-byte send/receive FIFOs. Besides, the two Universal Serial Bus ports provide high-speed data communication between peripherals and PC.

The on-board flash ROM is used to make the BIOS update easier. The high precision Real Time Clock/Calendar is built to support Y2K for accurate scheduling and storing configuration information. All of these features make PPAP-2020 excellent in stand-alone applications.

If any of these items is damaged or missing, please contact your vendor and save all packing materials for future replacement and maintenance.

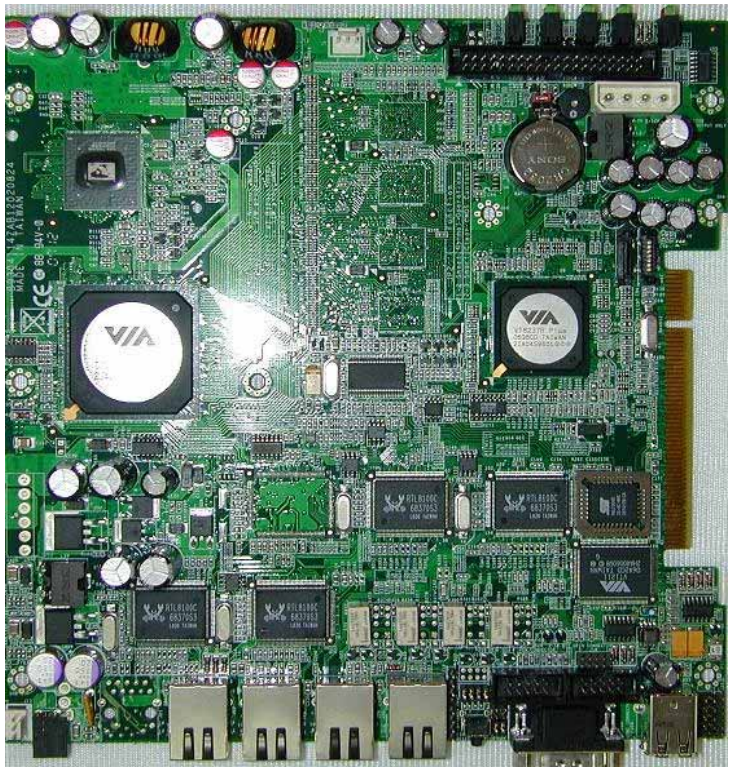


Figure 3-1 PPAP-2020 Board

3.2 System Architecture

The following illustration of block diagram will show you how PPAP-2011 gives you a highly integrated system solution. The most up-to-date system architecture of PPAP-2011 includes two main chips. It contains VIA CN700 and VIA VT823R+ to support VIA C7 processor, DDR2 SODIMM, USB 2.0 port, communication, Ultra DMA/100 IDE Master and SATA storage. The on-board super VIA VT8237R+ supports two UARTs and hardware monitoring.

PPAP-2020 has built-in onboard VIA C7 processor EBGA package (Eden C7) 533 or 400MHz system bus) for cost-effective and high performance application.

The VIA CN700 provides a completely integrated solution for the system controller and data path components in a VIA processor system. It provides optimized 64-bit DDR2 interface.

The VIA VT8237R+ provides a highly integrated multifunction for the best industry applications. It supports up to for Ultra ATA/33/66/100 IDE master interface, Universal Serial Bus **(USB2.0)** controllers,Full duplex high performance 150MB/s Dual Channel SATA interface.

All detailed operating relations are shown in ***Fig. 3-2*** (PPAP-2020 System Block Diagram).

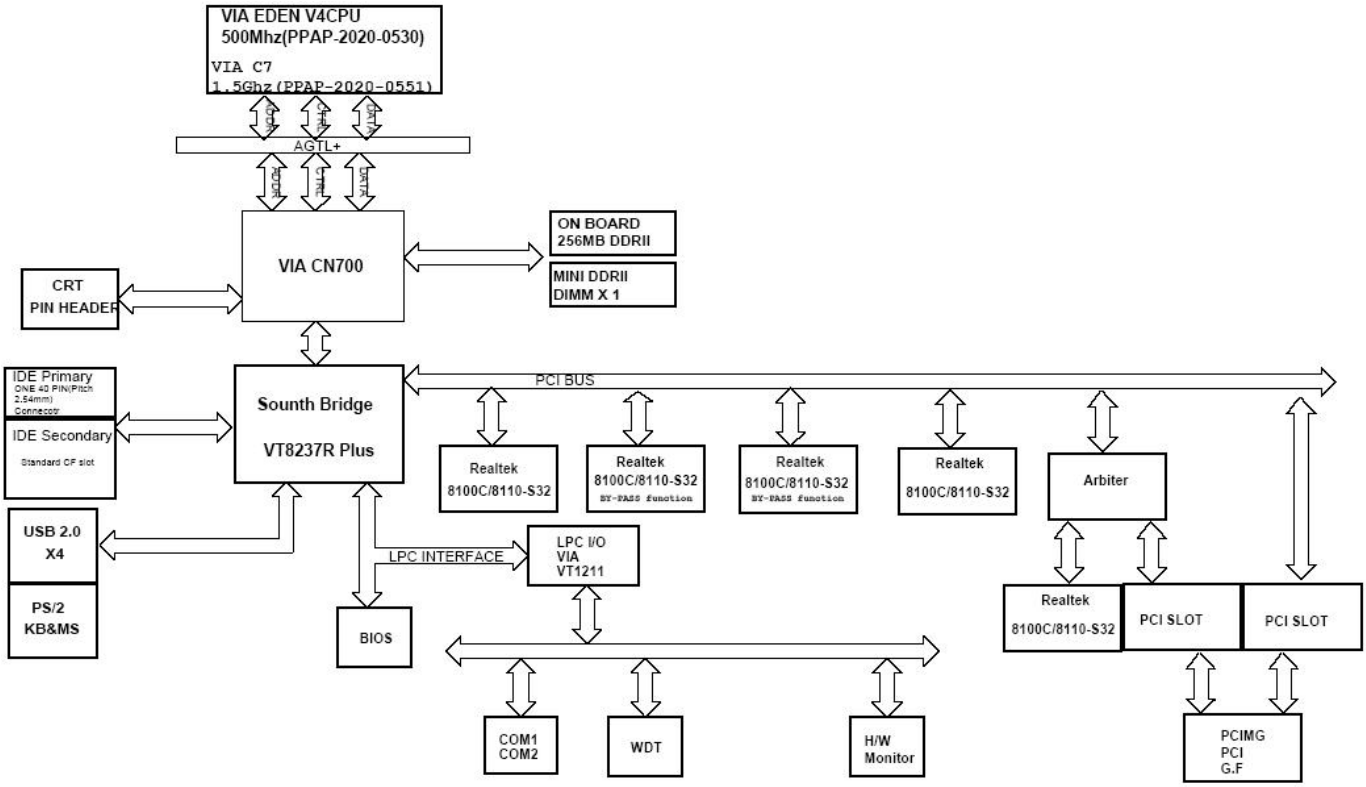


Figure 3-2 PPAP-2020 Block Diagram